



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS FISI


TESINA : COMPRESIÓN DE IMÁGENES
UTILIZANDO CLUSTERING BAJO EL
ENFOQUE DE COLONIA DE HORMIGAS.

AUTOR : YARASCA CARRANZA JUAN EDUARDO

ASESORA : RUIZ RIVERA MARIA ELENA

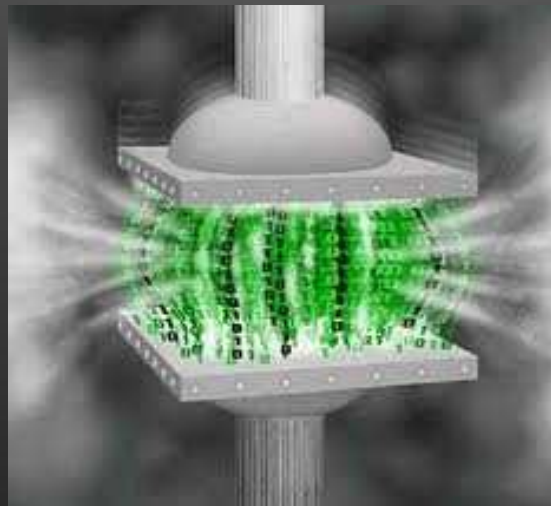
2013

Agenda

- ▶ Introducción
 - ▶ Objetivos
 - ▶ Marco Teórico
 - ▶ Estado del Arte
 - ▶ Método Propuesto
 - ▶ Experimentos Computacionales
 - ▶ Conclusiones
- 

Introducción

- ▶ Actualmente hay un continuo crecimiento en las telecomunicaciones por tal motivo se está dando lugar a un fenómeno de gran relevancia: el aumento en la demanda del tratamiento de imágenes, audio y video.
- ▶ A partir de la necesidad de transmitir y almacenar gran cantidad de información surge la idea de representarla de otra manera, de una forma más económica y sin alterar sustancialmente su significado, esto se logra mediante la compresión.



Introducción

▶ El Problema

- ▶ El continuo avance en la tecnología de las telecomunicaciones ha conllevado a un aumento en la demanda del uso de imágenes, audio y video en las diversas actividades cotidianas.
- ▶ Esto ha generado ciertos aspectos a tener en cuenta como el elevado tiempo empleado para la transmisión de datos y el alto costo de almacenamiento de este tipo de archivos
- ▶ El problema abordado en la investigación está vinculado al almacenamiento digital de imágenes.

Objetivos

- ▶ **Objetivo General**
- ▶ Diseñar e implementar un algoritmo que emplee el método de clustering bajo el enfoque de colonia de hormigas, con la finalidad de solucionar el problema de alto costo de almacenamiento digital de imágenes sin afectar drásticamente la visualización de las mismas.



Figura 1: Compresión basada en colonia de hormigas

Objetivos

▶ **Objetivos Específicos**

- ▶ Identificar algoritmos que permitan complementar al algoritmo de clustering de colonia de hormigas y mejoren la compresión de imágenes sin alterar la correcta visualización de las mismas.
- ▶ Obtener un radio de compresión entre 40% a 60% de tal manera que la imagen generada por el sistema de compresión sea más ligera que la imagen original.

Marco Teórico

- ▶ **La Imagen Digital**
- ▶ Una imagen digital es una representación bidimensional de una imagen utilizando bits [Ragone y Vittori, 2006] .

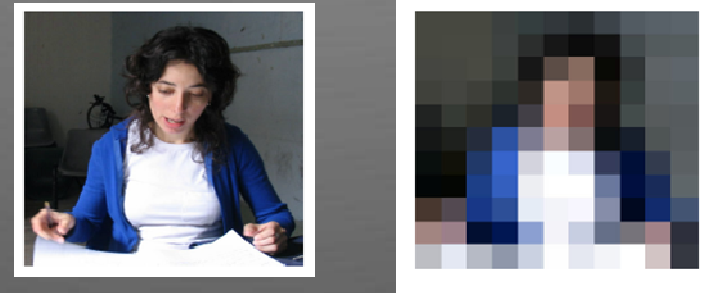


Figura 2: Imagen Digital

- ▶ **Píxel**
- ▶ Un píxel o pixel es la menor unidad homogénea en color que forma parte de una imagen digital [Ragone y Vittori, 2006] .

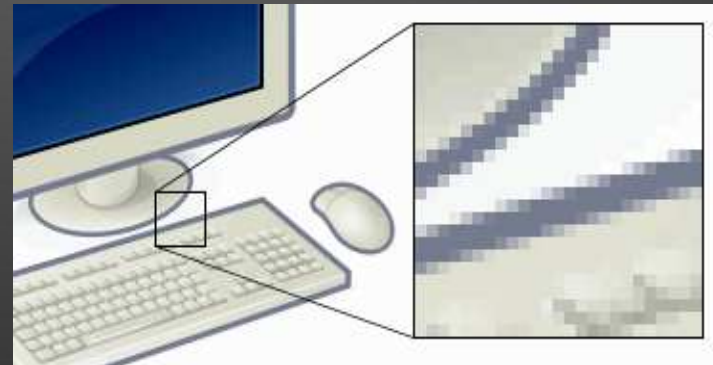


Figura 3: Píxeles de una imagen

Marco Teórico

- ▶ Tipos de Imágenes Digitales [Ragone y Vittori, 2006]

- ▶ 1.- Mapa de Bits

- ▶ Son las más usadas en la fotografía digital. Están formados por una rejilla de celdas a modo de mosaico a las cuales se les asigna un valor de color y luminancia.

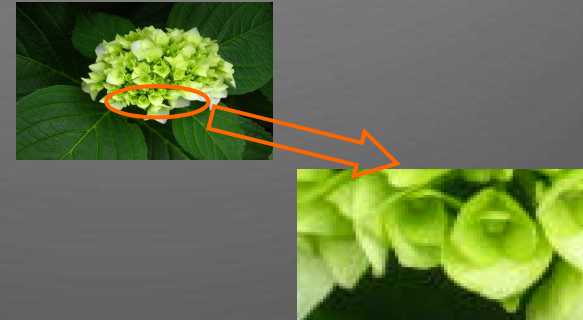


Figura 4: Mapa de Bits

- ▶ 2.- Imágenes Vectoriales

- ▶ Llamadas también gráficos vectoriales, son las más usadas en el dibujo por computadora. Están formadas por fórmulas que representan objetos como curvas, figuras y colores.

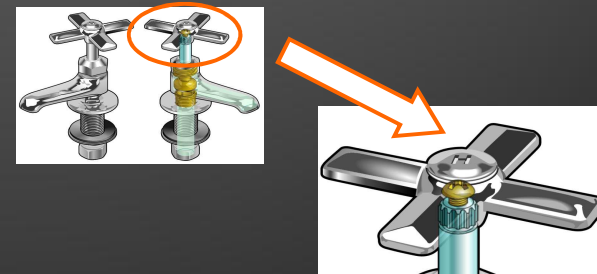


Figura 5: Imagen Vectorial

Marco Teórico

- ▶ **Profundidad del Color**
- ▶ Cantidad de luminosidad que lleva cada píxel de la imagen. [Ragone y Vittori, 2006] .
- ▶ Las profundidades de color más usuales:
 - ▶ 1 bit para imágenes de línea, sólo blanco y negro sin grises medios.
 - 8 bits para imágenes de tono continuo en blanco y negro, con 256 matices.
 - ▶ 8 bits para gráficos e iconos Web de color indexado
 - 24 bits (8 por cada canal, RGB) para imágenes a todo color.
 - 32 bits (8 por canal, CMYK) para archivos de impresión.

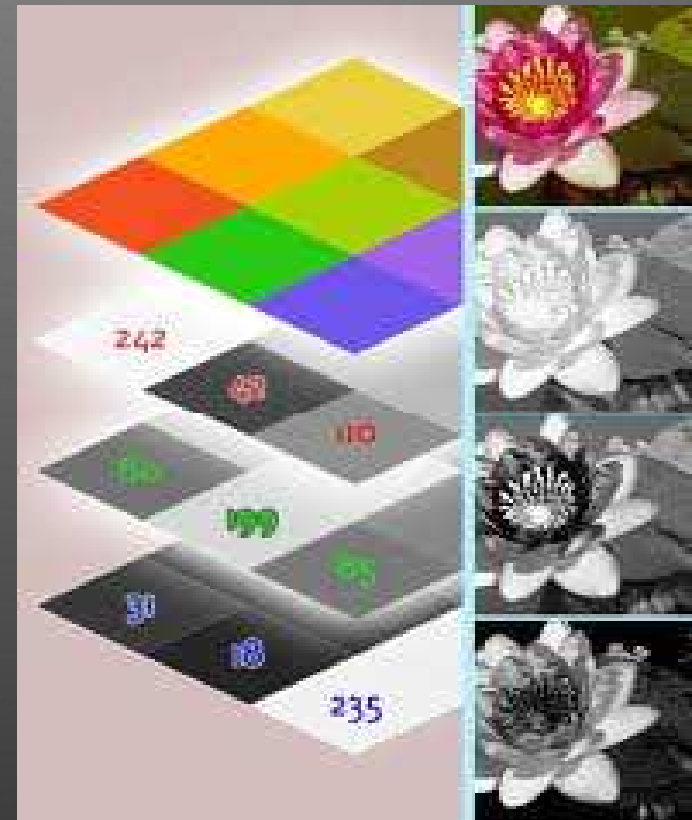


Figura 6: Profundidad del Color

Marco Teórico

► Almacenamiento de Datos [Seagate, 2008]

Número de archivos que un disco de 500 GB puede contener				
Tipo de medios	Aplicación o dispositivo	Archivo digital típico	Tamaño de archivo común*	La unidad de 500 GB almacena aprox. **
Películas y vídeo	Contenido multimedia de alta definición	Película de alta definición de 2 horas	10 GB	50 películas
		Programa de alta definición de 1 hora	5 GB	100 programas
	Contenido multimedia de definición estándar	Película de definición estándar de 2 horas	2,25 GB	222 películas
		Programa de definición estándar de 1 hora	1,12 GB	446 programas
	Videocámara	1 hora de vídeo digital RAW sin comprimir	12 GB	42 vídeos
Música	Reproductores de audio digitales	Canción de 4 minutos (128 Kbps)	4 MB	125.000 canciones
		Audio sin pérdida (WMA / FLAC) (~1.100 Kbps)	25 MB	20.000 canciones
		Sonido 5.1 Super HD (11.000 Kbps, sonido envolvente de 6 canales)†	230 MB	2.174 canciones
		Programa de audio de 30 minutos (64 Kbps)	15 MB	33.333 programas
Fotografías	Cámara digital	Fotografía de 4 megapíxeles‡	2 MB	250.000 fotografías
	Teléfonos móviles	Fotografía de teléfono móvil	0,4 MB	1.250.000 fotografías

*Los tamaños de los archivos (en decimales) representan el tamaño medio de este tipo de archivos.
 **El número de los archivos que puede contener un disco duro depende de que la unidad sólo contenga archivos de ese tipo.
 †Sonido envolvente digital MusicGigants
 ‡Cámara digital Canon

Figura 7: Capacidad de Almacenamiento

Marco Teórico

- ▶ **Compresión**
- ▶ La compresión de datos trata de reducir el número de bits necesarios para representar la información [Piñeiro, 2010].



Figura 8: Compresión

- ▶ **Clustering**
- ▶ Proceso de organizar objetos en grupos y cuyos elementos sean similares de alguna manera [Matteucci, 2008].

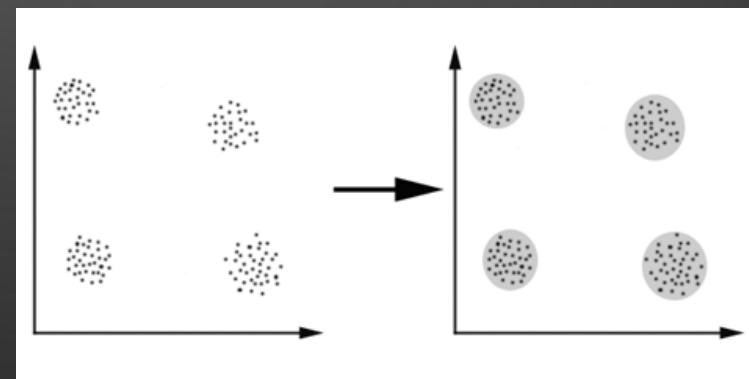


Figura 9: Clustering

Marco Teórico

- ▶ **Cuantificación Vectorial**
- ▶ Representar un vector N-dimensional que puede tomar cualquier valor continuo en cada una de sus N componentes mediante otro vector seleccionado de entre un conjunto finito de vectores N-dimensionales (Codebook) [Galindo, García, Barrientos, 1999].

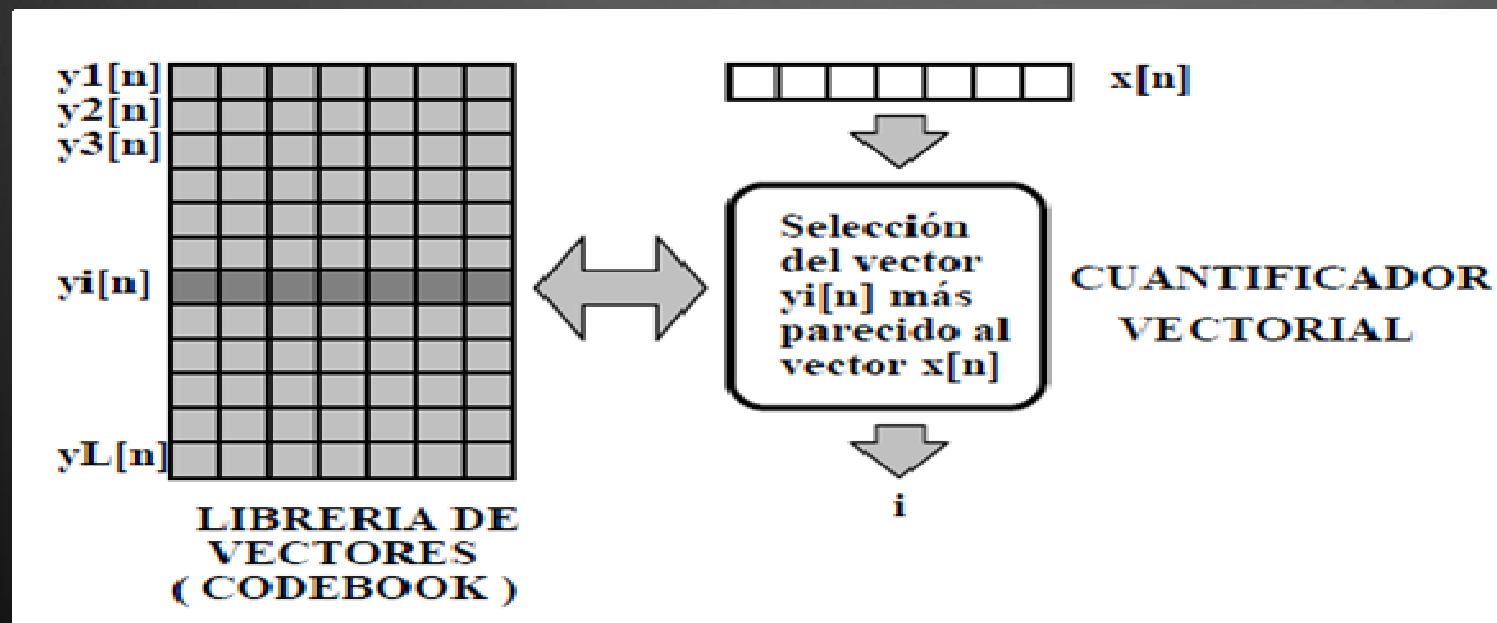


Figura 10: Cuantificación Vectorial

Marco Teórico

- ▶ **Algoritmo de Colonia de Hormigas**
- ▶ Es un algoritmo para encontrar las rutas óptimas que se basa en el comportamiento de las hormigas en busca de alimento [Viktor Macura, Ant Colony Algorithm].

Tiene aplicaciones mencionadas por Dorigo y Stützle [Dorigo y Stützle, 2005] como :

- ❑ Problema del agente viajero
- ❑ Problema de asignación cuadrática
- ❑ Problemas de horarios
- ❑ Enrutamiento de vehículos
- ❑ Enrutamiento de redes orientadas a conexión
- ❑ Coloración de grafos
- ❑ Asignación de frecuencias

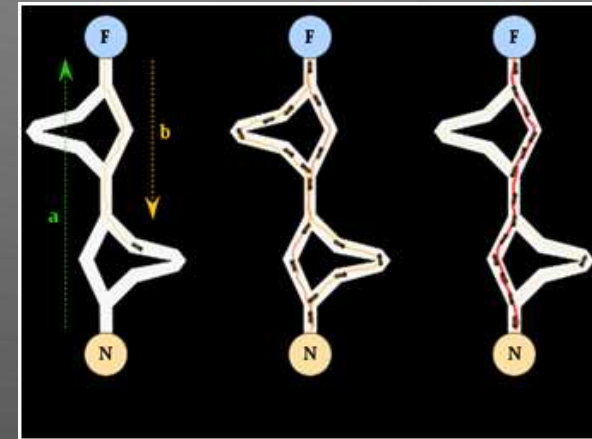


Figura 11: Algoritmo Colonia de Hormigas

Marco Teórico

- ▶ **Image Dithering**
- ▶ Image Dithering es una técnica usada para crear profundidades de color en imágenes que aparecen en dispositivos con paletas de colores limitadas [Cosma, 2004].



Figura 12: Image Dithering

Estado del Arte

- ▶ Según la clasificación establecida por la ACM (Association for Computing Machinery), el tema a desarrollar en la tesina se encuentra en:

Reconocimiento de Patrones

Clustering

Algoritmos

Clustering basado en Colonia
de Hormigas



Estado del Arte

- ▶ La mayoría de las técnicas estándares tratan la compresión del color en las imágenes como un problema de agrupamiento (clustering) de puntos en el espacio tridimensional, donde los puntos representan los colores encontrados en la imagen original y los tres ejes son representados por los tres canales de color (rojo, verde y azul).
- ▶ Algunos algoritmos usados en la cuantificación de imágenes son los siguientes:
 - ▶ Algoritmo Median Cut.
 - ▶ Algoritmo NeuQuant.
 - ▶ Algoritmo Local K – Means.
- ▶ Otros algoritmos utilizados:
 - ▶ Algoritmo Spatial Color Quantization
 - ▶ Algoritmo Uniforme
 - ▶ Algoritmo Popularity
 - ▶ Algoritmo Octree

Estado del Arte

- ▶ **I.- Algoritmo Median Cut**
- ▶ El algoritmo Median Cut fue presentado por Paul Heckbert en su artículo “Color Image Quantization for Frame Buffer Display” en 1982 [Heckbert, 1982].

En este método se pueden distinguir 4 fases en la cuantificación de imágenes:

1. El muestreo de la imagen original para las estadísticas del color.
2. La elección de un mapa de colores (colourmap) basados en las estadísticas del color.
3. Mapear los colores originales hacia sus vecinos más cercanos en el mapa de colores.
4. Cuantificación y redibujo de la imagen.

Las imágenes utilizando el algoritmo Median Cut empleando 16, 8, 4 y 2 bits / píxel [Mota et al., 2001].



Figura 13: Median Cut

Estado del Arte

► II.- Algoritmo NeuQuant

El algoritmo NeuQuant fue presentado por Anthony Dekker en su artículo “Kohonen Neural Networks for Optimal Colour Quantization” en 1994 [Dekker, 1994].

El algoritmo funciona con una dimensión de auto-organización de redes neuronales de Kohonen, por lo general con 256 neuronas, que se auto-organizan a través del aprendizaje para que coincidan con la distribución de los colores de la imagen de entrada.

Durante la fase de aprendizaje se puede utilizar diversos factores de muestreo. Cada neurona toma la posición en el espacio RGB lo cual genera un mapa a color de alta calidad.

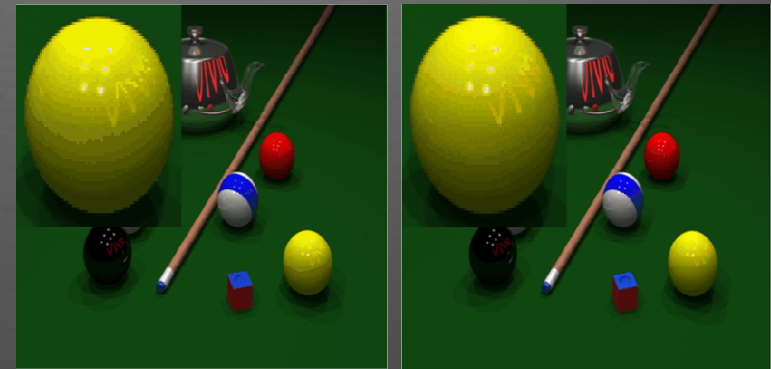


Figura 14: NeuQuant

Estado del Arte

- ▶ **III.– Algoritmo Local K– Means**
- ▶ El algoritmo fue propuesto por Stuart Lloyd en 1957 como una técnica de la Modulación Código – Pulso, pero no fue publicado hasta 1982, fecha en la que presentó su artículo "Least square quantization in PCM". Este algoritmo también es conocido como el algoritmo de Lloyd [Lloyd, 1982].
- ▶ El algoritmo sigue los siguientes pasos:
 - ▶ 1.– Elegir las k – medias (means) iniciales en forma aleatoria.
 - ▶ 2.– Se crean k clusters asociando cada observación a la media más cercana, dichas particiones pueden representarse mediante el diagrama de Voronoi.
 - ▶ 3.– El centroide de cada cluster se convierte en la nueva media.
 - ▶ 4.– Se repiten los pasos 2 y 3 hasta que se alcance la convergencia.

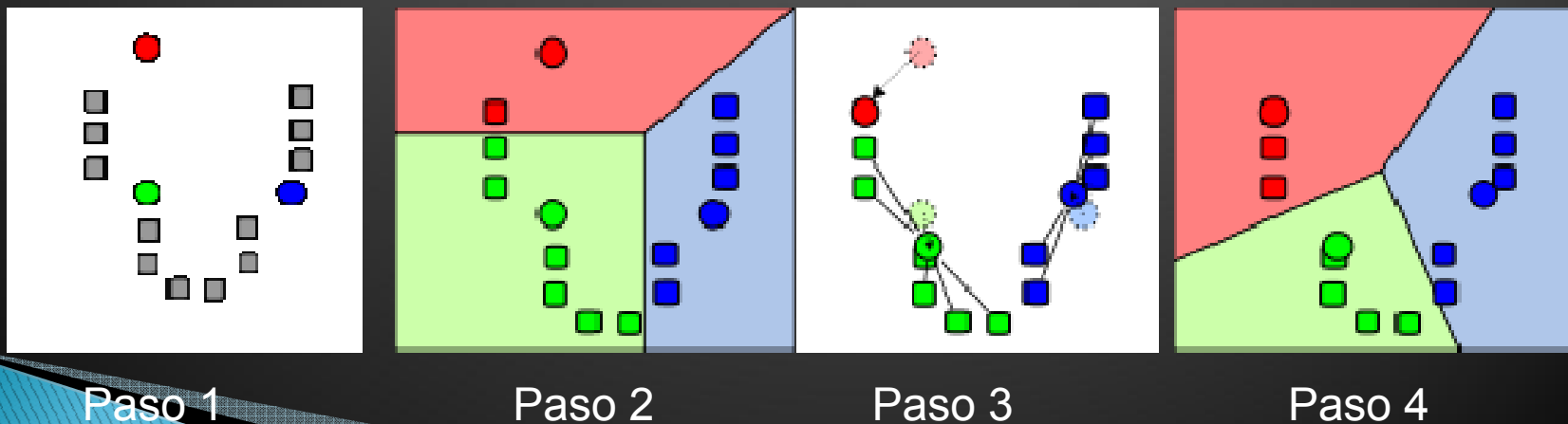


Figura 15: K – Means

Estado del Arte

- ▶ **IV.- Ant - Based Clustering**
- ▶ Ant - Based Clustering es un método heurístico de clustering y ordenamiento inspirado en el comportamiento de las hormigas en la naturaleza. El primer algoritmo Ant - Based Clustering fue propuesto por Deneubourg en su artículo "The dynamics of collective sorting: Robot-like ants and Ant-like robots" en 1991 [Deneubourg et al., 1991] para producir comportamiento de clustering en grupos de robots.

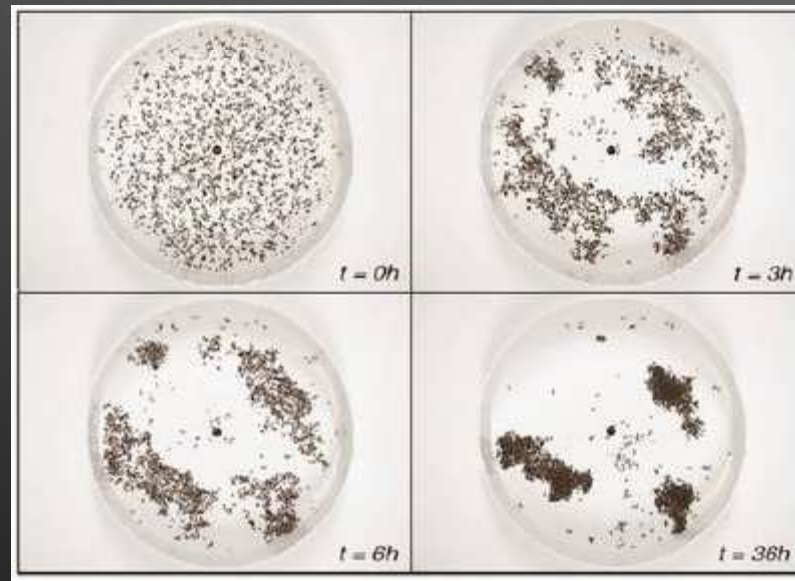


Figura 16: Ordenamiento de Hormigas

Estado del Arte

- ▶ IV.- Ant - Based Clustering
- ▶ Deneubourg presenta funciones en donde se definen las funciones de probabilidad de recoger o soltar un ítem.

$$p_{pick}(i) = \left(\frac{k^+}{k^+ + f(i)} \right)^2$$

$$p_{drop}(i) = \left(\frac{f(i)}{k^- + f(i)} \right)^2$$

De donde k^+ y k^- que determinan la influencia de las funciones vecindario $f(i)$, la cual es una estimación de la fracción de datos en el ambiente inmediato de la hormiga y que dependiendo de la similitud que exista la hormiga recogerá o soltará el ítem.

Estado del Arte

▶ IV.- Ant - Based Clustering

- ▶ La limitación de la memoria a corto plazo de las hormigas fue superada por Lumer y Faieta [Lumer y Faieta, 1994], donde consideran el promedio de similitud entre el ítem i con todos los ítems j dentro de su vecindad. La función $f(i)$ es calculada mediante:

$$f(i) = \max \left(0, \frac{1}{\sigma^2} \sum_j \left(1 - \frac{d(i,j)}{\alpha} \right) \right)$$

En donde

$d(i,j) \in [0,1]$ es la función de similitud definida para 2 puntos de data en el espacio, medida del coseno.

$\alpha \in [0,1]$ es un parámetro de escalamiento dato - dependiente,

σ^2 es el tamaño del vecindario local, generalmente $\sigma^2 \in [9, 25]$.

El agente es colocado en el centro de su vecindario local y cuyo radio de percepción en todas las direcciones es $(\sigma-1)/2$.

Estado del Arte

▶ IV.– Ant – Based Clustering

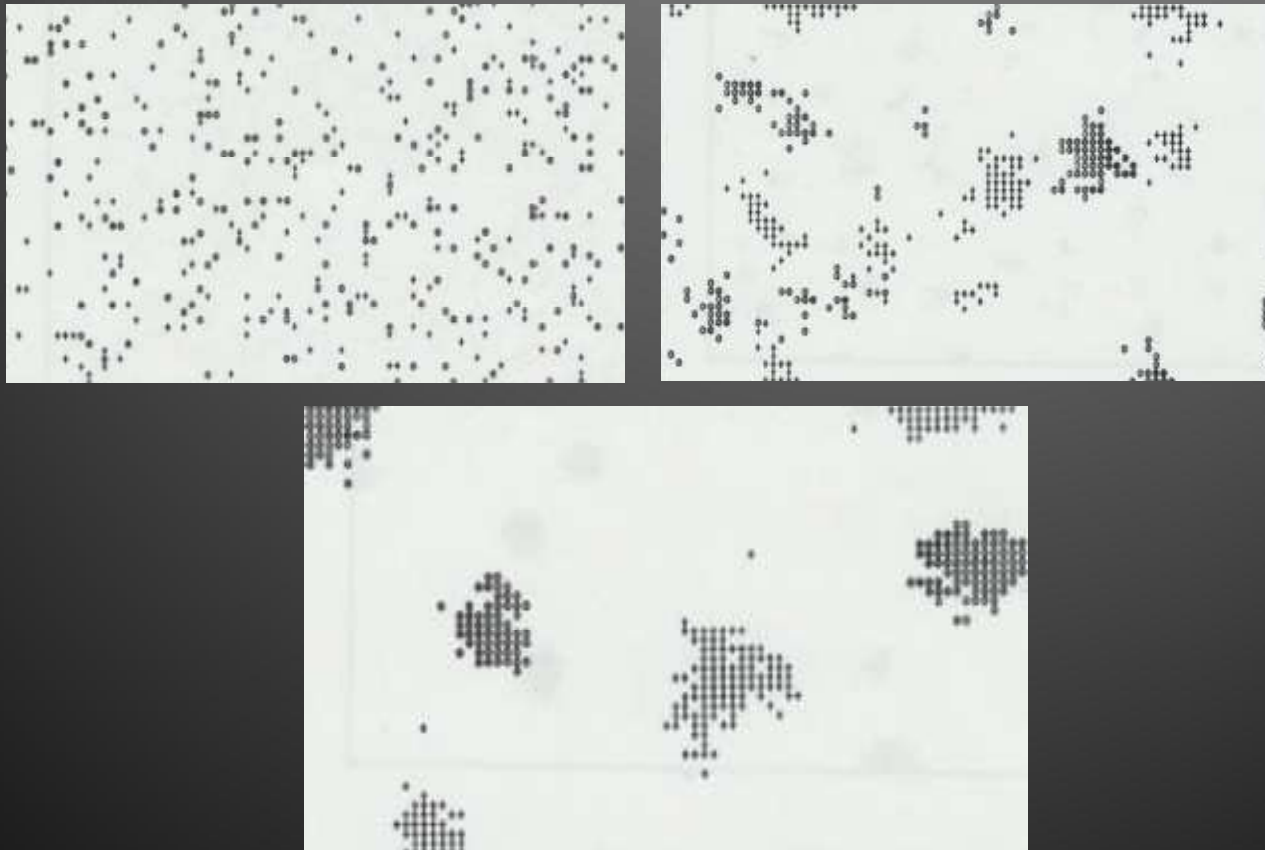


Figura 17: Ant – Based Clustering

Estado del Arte

- ▶ **V.– Casos de Éxito del Algoritmo de Colonia de Hormigas**
- ▶ **Desarrollo de una aplicación para la gestión, clasificación y agrupamiento de documentos económicos con algoritmos bio-inspirados [Cobo y Rocha, 2007]**, este trabajo fue desarrollado por Ángel Cobo y Rocío Rocha, en el cual desarrollan una aplicación Web que utiliza técnicas bio-inspiradas para clasificar y agrupar colecciones multilingües de documentos en el campo de la economía y los negocios.
- ▶ **Enrutamiento Multicast Multiobjetivos basado en Colonia de Hormigas [Pinto et al., 2005]**, desarrollado por Diego Pinto, Hugo Estigarribia, y Benjamín Barán. Proponen un algoritmo multiobjetivo, para la resolución del problema de Enrutamiento Multicast en Ingeniería de Tráfico, que está basado en Ant Colony Optimization y que es utilizado para la construcción del árbol multicast, en el contexto de transmisión de datos en redes de computadoras.
- ▶ **Selección de rutas de distribución utilizando optimización por colonia de hormigas [Feitón y Cespón, 2009]**, trabajo desarrollado por Michael Feitó y Roberto Cespón, en donde se enfocan en el diseño y aplicación de un procedimiento metaheurístico para la selección de rutas de distribución, que permita la reducción de los costos de transporte y un mejor tiempo de entrega a partir de la optimización de las distancias a recorrer.

Estado del Arte

- ▶ V.– Algoritmos utilizados en Dithering

- ▶ En el artículo desarrollado por Ovidiu Cosma [Cosma, 2004] menciona 3 tipos de algoritmos de dithering:
 - ▶ Noise Dithering
 - ▶ Ordered Dithering
 - ▶ Error Difussion

- ▶ 1.– Noise Dithering

En el cual para cada punto en la escala de grises de una imagen genera un número aleatorio entre 0 y 255, si el número generado es mayor que el valor de la imagen en ese punto entonces se pintará de color blanco de lo contrario será de color negro [Croker et al., 1989].



Figura 18: Noise Dithering

Estado del Arte

- ▶ V.– Algoritmos utilizados en Dithering
- ▶ 2.– Ordered Dithering

La imagen es representada con un conjunto de patrones predefinidos [Cosma, 2004].

El algoritmo aplica un mapa de umbrales llamado Matriz de Bayer o Matriz de Índices sobre cada píxel de la imagen, causando que algunos de los píxeles sean pintados con un color diferente, dependiendo de que tan lejos se encuentre el color generado del color de la paleta de colores disponible [Crocker et al., 1989].

$$\frac{1}{5} \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix} \quad \frac{1}{10} \begin{bmatrix} 3 & 7 & 4 \\ 6 & 1 & 9 \\ 2 & 8 & 5 \end{bmatrix}$$

Figura 18: Matriz de Bayer



Figura 19: Ordered Dithering

Estado del Arte

- ▶ V.– Algoritmos utilizados en Dithering
- ▶ 3.– Error Diffusion

La cuantificación residual se distribuye en los píxeles que aún no han sido procesados produciendo mejores resultados que los algoritmos de noise y ordered dithering [Cosma, 2004].

El algoritmo Floyd–Steinberg encuentra para cada punto de la imagen el más cercano en intensidad de color disponible. A continuación, calcula la diferencia entre el valor de la imagen en ese punto y el valor más cercano encontrado, valor de error. Luego divide el valor de error y lo distribuye a algunos de los píxeles vecinos que no han sido procesados, estos nuevos valores, se convierten en los valores de los píxeles que se utilizarán para su procesamiento

$\frac{1}{16}$	0	0	0
	0	0	7
	3	5	1

Figura 20: Matriz de Distribución del Error



Figura 20: Error – Diffusion Dithering

Estado del Arte

ALGORITMO	VENTAJAS	DESVENTAJAS
Median Cut	<ul style="list-style-type: none"> - Simplicidad de implementación. - Poco uso de recursos de memoria y procesador de la PC. 	<ul style="list-style-type: none"> - Soluciones poco óptimas. - Generación de codebook limitada. - No cuenta con procesamiento en paralelo.
NeuQuant	<ul style="list-style-type: none"> - Aprendizaje adaptativo. - Tolerancia a fallos. - Dinámico - Procesamiento paralelo 	<ul style="list-style-type: none"> - Necesidad de un tiempo para el entrenamiento de la red neuronal. - Gran coste de recursos de memoria y procesador de la PC. - Implementación complicada.
K-Means	<ul style="list-style-type: none"> - Simplicidad de implementación. - Técnica muy usada en la identificación de patrones. 	<ul style="list-style-type: none"> - Excesivo tiempo para llegar a la solución final cuando se maneja gran cantidad de data. - Posibilidad de la no convergencia del método. - No cuenta con procesamiento en paralelo
Ant Based Clustering	<ul style="list-style-type: none"> - Procesamiento paralelo. - Mejora constante del algoritmo. - Auto-organización - Algoritmo robusto - No necesitan información a priori. 	<ul style="list-style-type: none"> - Complejidad de implementación media.

Estado del Arte

► Comparación de Algoritmos de Clustering

Ítem	Algoritmos											
	Median Cut			K – Means			NeuQuant			Ant – Based Clustering		
	Peso	Valor	Ponderado	Peso	Valor	Ponderado	Peso	Valor	Ponderado	Peso	Valor	Ponderado
Robustez	0.35	1	0.35	0.35	2	0.70	0.35	2	0.70	0.35	3	1.05
Paralelismo	0.30	1	0.30	0.30	1	0.30	0.30	3	0.90	0.30	3	0.90
Eficacia	0.35	1	0.35	0.35	2	0.70	0.35	3	1.05	0.35	3	1.05
Totales	1.00	3	1.00	1.00	5	1.70	1.00	8	2.65	1.00	9	3.00

Leyenda:

Descripción	Valor
Baja	1
Media	2
Alta	3

Estado del Arte

► Comparación de Algoritmos de Clustering

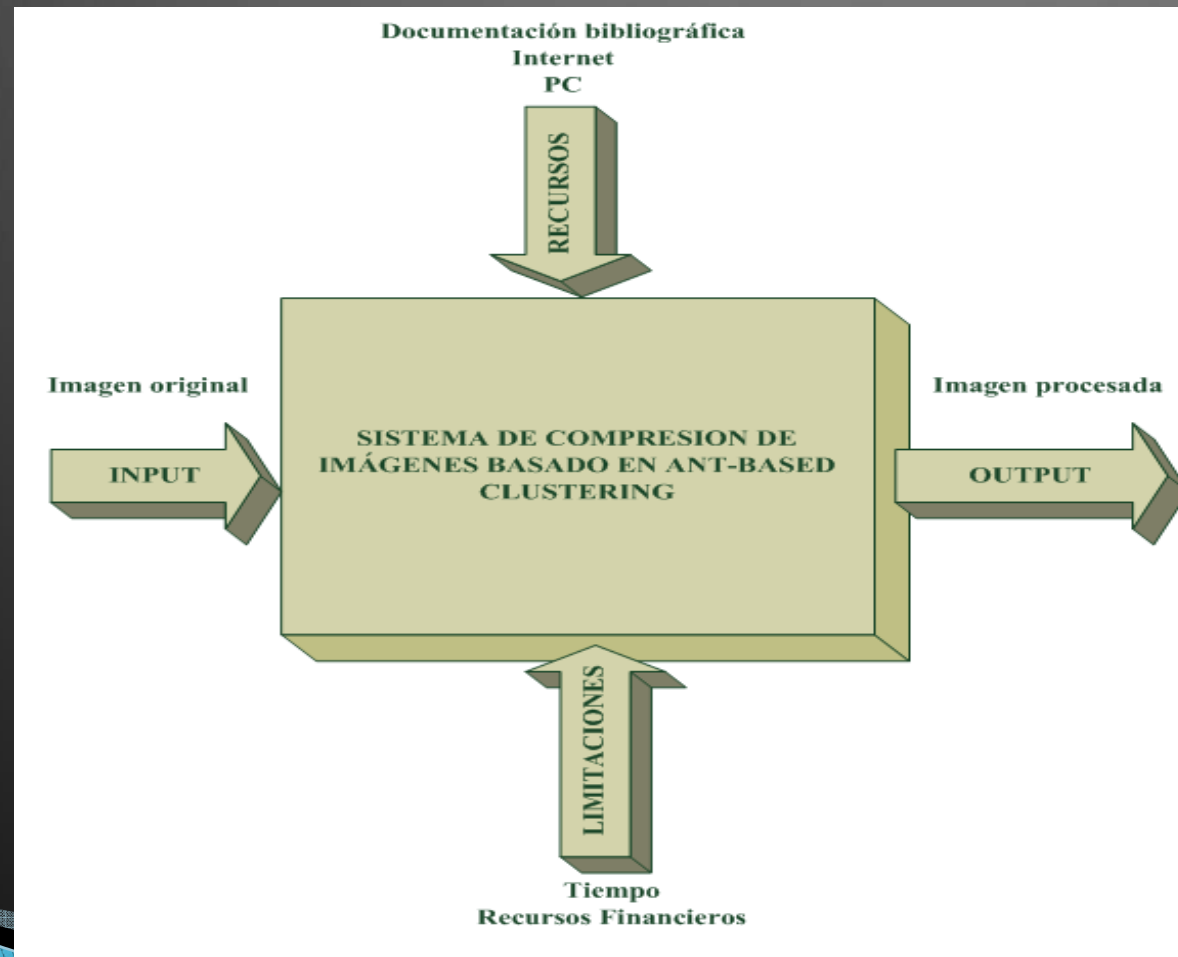
Ítem	Algoritmos											
	Median Cut			K – Means			NeuQuant			Ant – Based Clustering		
	Peso	Valor	Ponderado	Peso	Valor	Ponderado	Peso	Valor	Ponderado	Peso	Valor	Ponderado
Complejidad de implementación	0.3	2	0.6	0.3	2	0.6	0.3	3	0.3	0.3	2	0.6
Uso de recursos computacionales	0.3	2	0.6	0.3	2	0.6	0.3	3	0.9	0.3	2	0.6
Tiempo de procesamiento	0.3	2	0.6	0.3	2	0.6	0.3	3	0.9	0.3	2	0.6
Necesidad de información a priori	0.1	1	0.1	0.1	1	0.1	0.1	3	0.1	0.1	1	0.1
Totales	1.0	12	1.9	1.0	12	1.9	1.0	10	2.2	1.0	12	1.9

Leyenda:

Descripción	Valor
Baja	1
Media	2
Alta	3

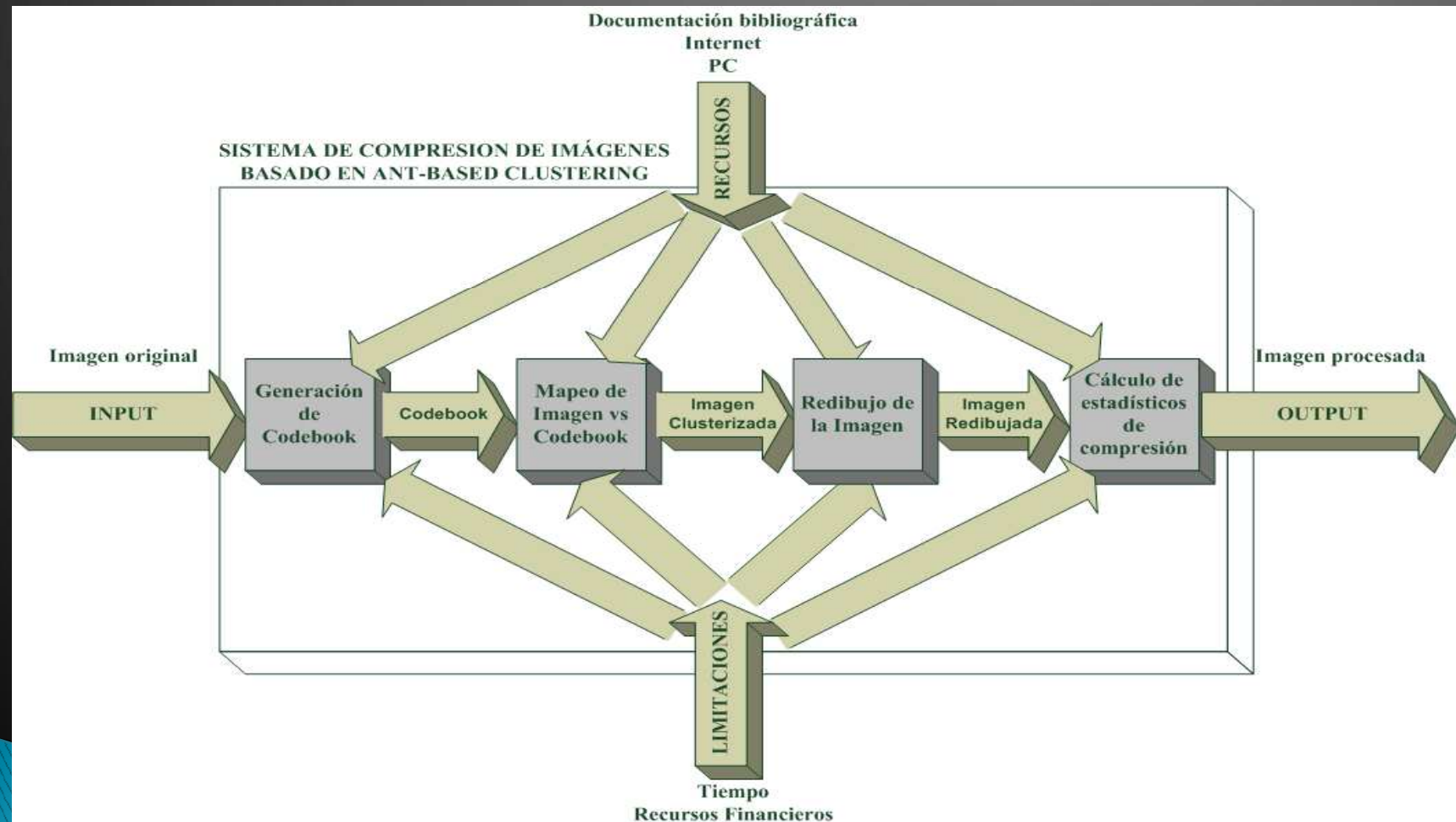
Método Propuesto

- I.- Diagrama de Caja Negra



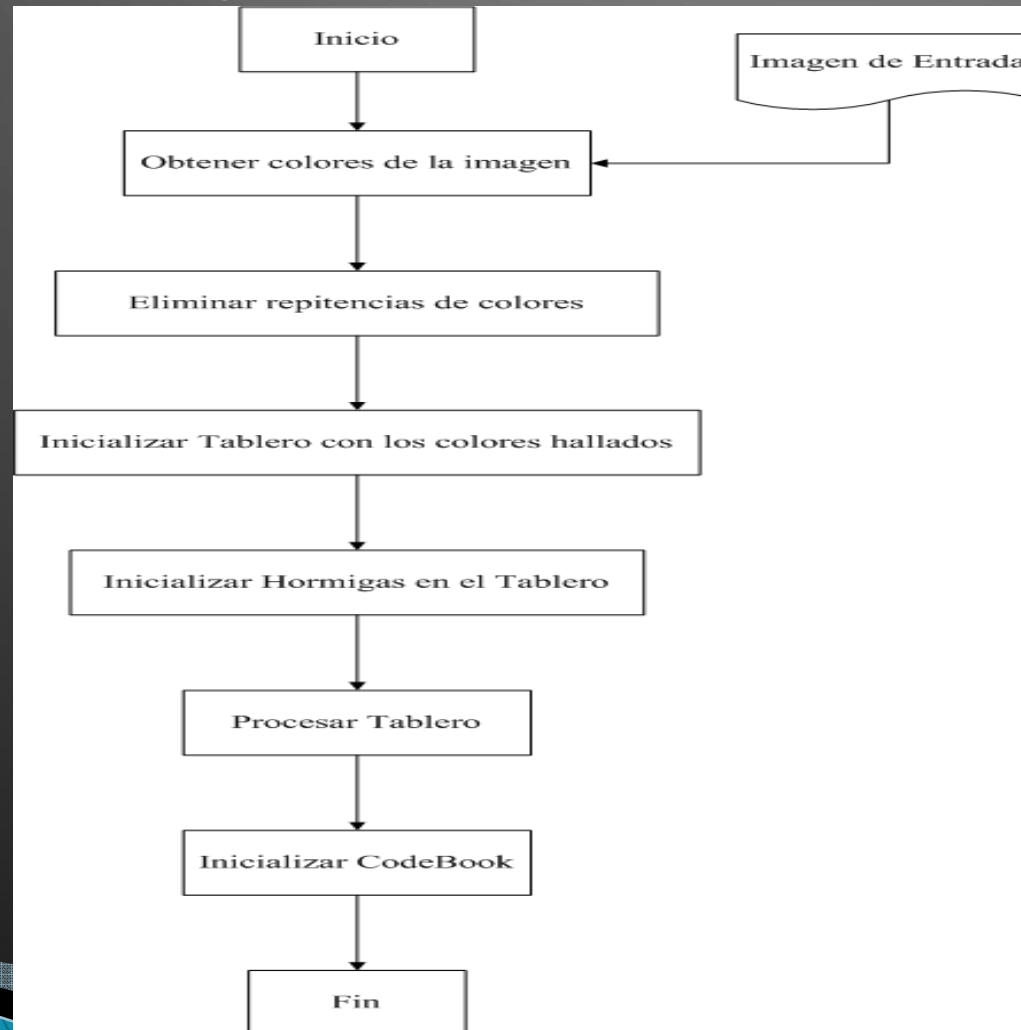
Método Propuesto

► II.- Diagrama de Caja Blanca



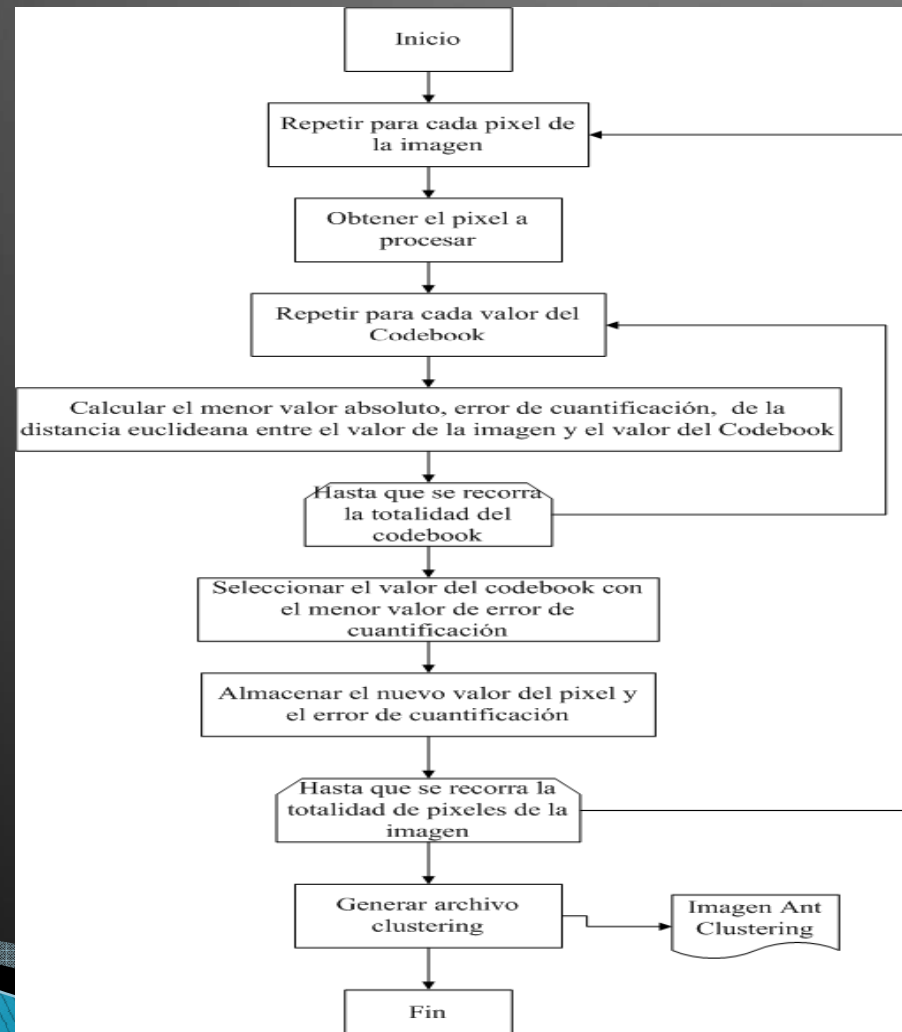
Método Propuesto

► II.1.– Generación de Codebook



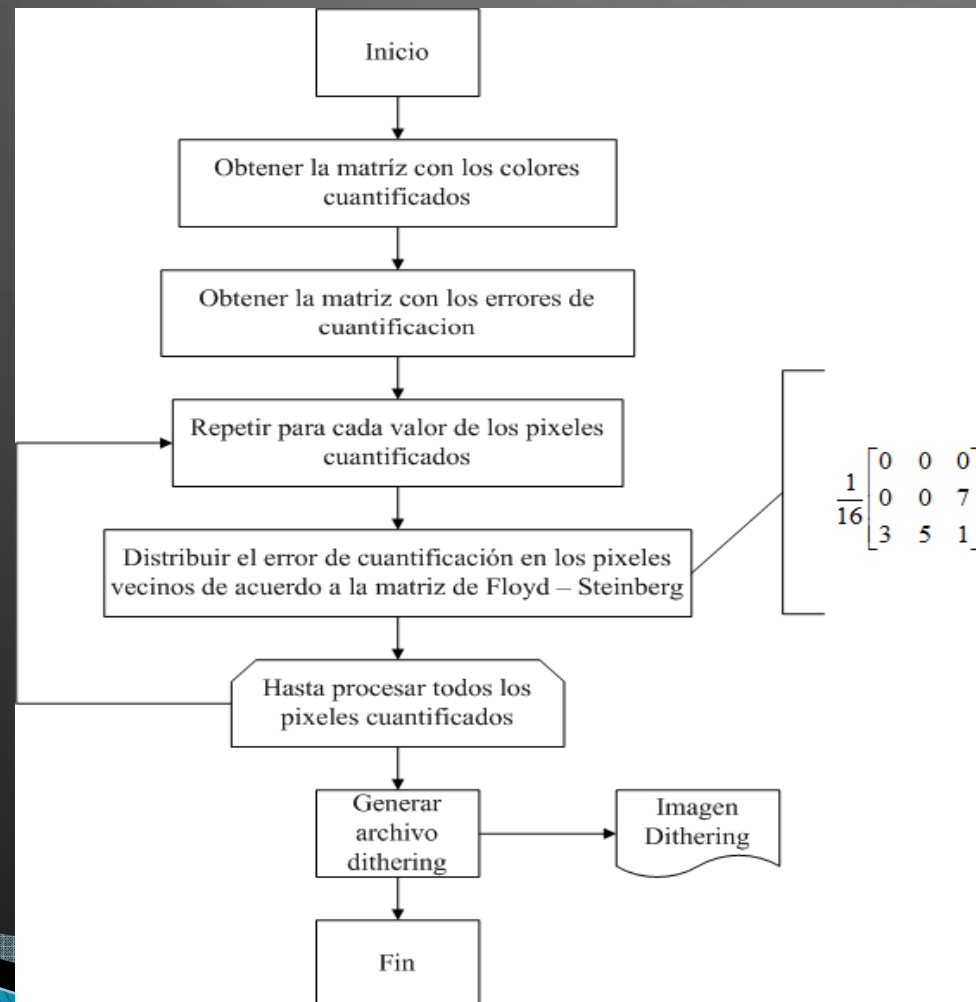
Método Propuesto

► II.2.– Mapeo de Imagen vs Codebook



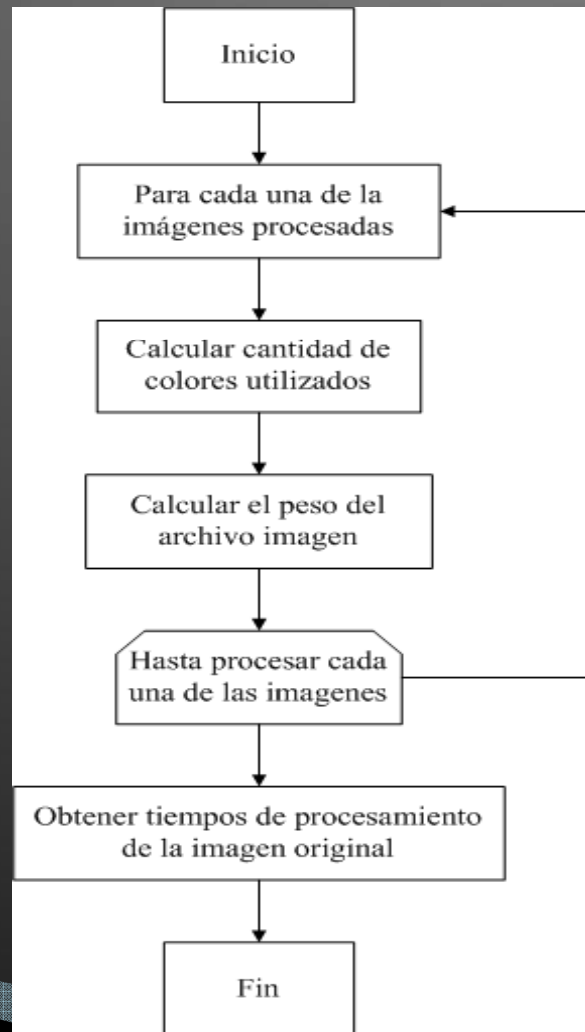
Método Propuesto

► II.3.– Redibujo de la Imagen



Método Propuesto

► II.4.– Cálculo de Estadísticos de Compresión



Método Propuesto

► III.– Prototipo de Interface Gráfica de Usuario

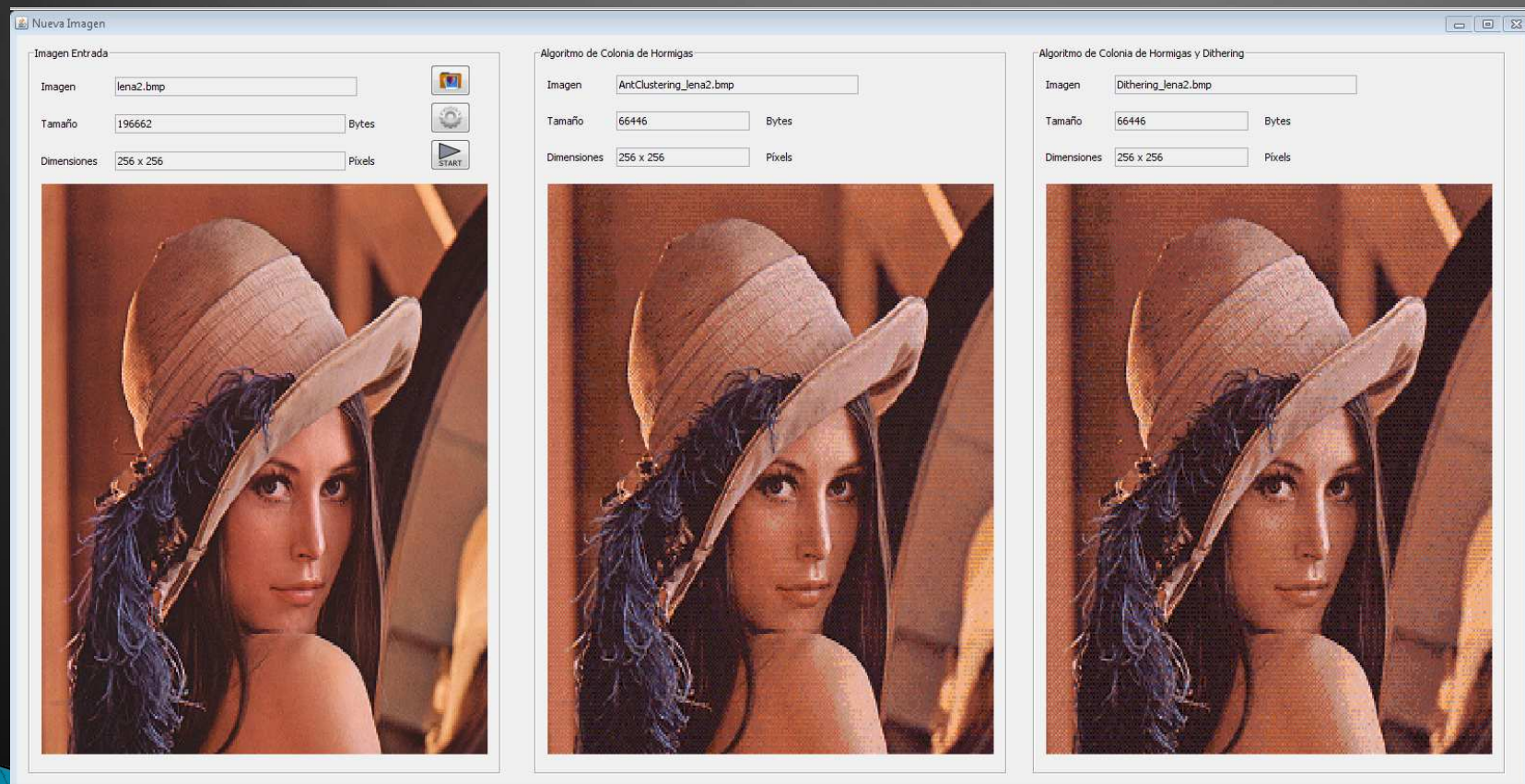


Figura 21: Prototipo de Interface de Usuario

Experimentos Computacionales

► I.- Pruebas del Sistema

- Longitud de Avance de las Hormigas = 5
- Radio de Vecindad del Clúster = 30
- Cantidad de Hormigas = 20

Imagen	Dimensiones de la Imagen	Imagen Original		Imagen Clustering		Imagen Dithering	
		Nº Colores	Peso (Bytes)	Nº Colores	Peso (Bytes)	Nº Colores	Peso (Bytes)
Lena	256 x 256	44664	196662	100	65990	57	65990
Tree	256 x 256	44380	196662	155	66210	76	66210
Airplane	256 x 256	27732	196662	87	65938	48	65938
Auto	604 x 453	16626	820890	111	274110	56	274110
Couple	256 x 256	26527	196662	192	65966	74	65966
Girl 1	256 x 256	40879	196662	139	66146	73	66146
Girl 2	256 x 256	15756	196662	79	65906	41	65906
Girl 3	256 x 256	32485	196662	94	65966	44	65966
House	256 x 256	48617	196662	68	65862	38	65862
House 2	256 x 256	33925	196662	101	65994	40	65994
Jelly beans 1	256 x 256	18646	196662	193	66362	82	66362
Jelly beans 2	256 x 256	25867	196662	102	65998	45	65998
León Aguila	796 x 421	78252	1005402	132	335698	68	335698
Peppers	256 x 256	55451	196662	116	66054	47	66054
Babbon	256 x 256	61770	196662	164	66246	87	66246
Sailboat on Lake	256 x 256	51954	196662	211	66434	74	66434
Splash	256 x 256	37736	196662	171	66274	89	66274
Tiffany	256 x 256	31893	196662	187	66338	60	66338
Juan	536 x 480	56190	771894	142	257902	61	257902

Experimentos Computacionales

- ▶ I.- Pruebas del Sistema



Figura 22: Imagen Lena

Experimentos Computacionales

- ▶ I.- Pruebas del Sistema



Figura 23: Imagen Tree

Experimentos Computacionales

- ▶ I.- Pruebas del Sistema



Figura 23: Imagen House

Experimentos Computacionales

- ▶ I.- Pruebas del Sistema



Figura 24: Imagen Babbon

Experimentos Computacionales

- ▶ I.- Pruebas del Sistema



Figura 25: Imagen Tiffany

Experimentos Computacionales

- ▶ I.- Pruebas del Sistema



Figura 24: Imagen Juan

Experimentos Computacionales

► II.- Comparación del Sistema Ant-Colony y MS Paint

Imagen	Dimensiones de la Imagen	Imagen Original	Sistema Ant – Colony Clustering		MS Paint	
		Peso (Bytes)	Peso (Bytes)	Porcentaje Compresión	Peso (Bytes)	Porcentaje Compresión
Lena	256 x 256	196662	65990	66.44	66614	66.13
Tree	256 x 256	196662	66210	66.33	66614	66.13
Airplane	256 x 256	196662	65938	66.47	66614	66.13
Auto	604 x 453	820890	274110	66.61	274690	66.54
Couple	256 x 256	196662	65966	66.26	66614	66.13
Girl 1	256 x 256	196662	66146	66.37	66614	66.13
Girl 2	256 x 256	196662	65906	66.49	66614	66.13
Girl 3	256 x 256	196662	65966	66.46	66614	66.13
House	256 x 256	196662	65862	66.51	66614	66.13
House 2	256 x 256	196662	65994	66.44	66614	66.13
Jelly beans 1	256 x 256	196662	66362	66.26	66614	66.13
Jelly beans 2	256 x 256	196662	65998	66.44	66614	66.13
León Águila	796 x 421	1005402	335698	66.61	336194	66.56
Peppers	256 x 256	196662	66054	66.41	66614	66.13
Babbon	256 x 256	196662	66246	66.31	66614	66.13
Sailboat on Lake	256 x 256	196662	66434	66.22	66614	66.13
Splash	256 x 256	196662	66274	66.30	66614	66.13
Tiffany	256 x 256	196662	66338	66.27	66614	66.13
Juan	536 x 480	771894	257902	66.59	258358	66.53
Compresión Promedio				66.41		66.19

Experimentos Computacionales

► II.- Comparación del Sistema Ant-Colony y MS Paint



Figura 25: Imagen Lena

Experimentos Computacionales

► II.- Comparación del Sistema Ant-Colony y MS Paint



Figura 26: Imagen Tiffany

Experimentos Computacionales

► II.- Comparación del Sistema Ant-Colony y MS Paint



Figura 24: Imagen Tree

Experimentos Computacionales

► II.- Comparación del Sistema Ant-Colony y MS Paint



Figura 24: Imagen Juan

Conclusiones

- ▶ El algoritmo de clustering diseñado bajo el enfoque de colonia de hormigas que se presentó en esta investigación permitió comprimir imágenes y de esta manera reducir su tamaño de su almacenamiento digital sin afectar excesivamente su visualización por parte de las personas.
- ▶ El uso del algoritmo de dithering error – difusión complementó al algoritmo de clustering basado en colonia de hormigas de tal manera que al combinarlos las imágenes procesadas no fueron afectadas drásticamente en su percepción visual.
- ▶ El algoritmo de compresión basado en colonia de hormigas que se presentado permitió reducir el peso de las imágenes en aproximadamente un 66.41%, lo que es equivalente a las dos terceras partes su tamaño de almacenamiento en disco.

Bibliografía

- ▶ [ACM, 1998]ACM Computing Classification System. Accesado el 09 de febrero del 2011 desde <http://www.acm.org/about/class/ccs98-html>
- ▶ [Cobo y Rocha, 2007]
- ▶ Ángel Cobo Ortega, Rocío Rocha Blanco. Desarrollo de una aplicación para la gestión, clasificación y agrupamiento de documentos económicos con algoritmos bio-inspirados. Universidad de Cantabria. Departamento de Matemática Aplicada y Ciencias de la Computación, Departamento de Administración de Empresas. España. 2007.
- ▶ [Cosma, 2004]
- ▶ Ovidiu Cosma. Image Dithering based on Wavelet Transform. Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics. Grecia. 2004.

Bibliografía

- ▶ [Croker et al., 1989]
- ▶ Lee Daniel Crocker , Paul Boulay and Mike Morra. Dither. 1989.
Accesado el 28 de febrero de 2011 desde
<http://www.efg2.com/Lab/Library/ImageProcessing/DHALF.TXT>
- ▶ [Dekker, 1994] Kohonen Neural Networks for Optimal Color Quantization. Anthony Dekker. 1994.
- ▶ [Deneubourg et al., 1991]
- ▶ J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, y L. Chrétien. The dynamics of collective sorting: Robot-like ants and Ant-like robots. Université Libre de Bruxelles, Bélgica; University of Bath, Reino Unido, 1991.

Bibliografía

- ▶ **[Dorigo y Stützle, 2005] The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. Marco Dorigo, Thomas Stützle . 2005. Pp 24.**
- ▶ **[Feitón y Cespón, 2009]**
- ▶ **Feitó C. Michael, Cespón C. Roberto. Vector, Vol. 4. Universidad Central "Marta Abreu" de Las Villas. Facultad de Ingeniería Industrial y Turismo. Cuba. 2009, Pp. 59 – 66.**
- ▶ **[Galindo, García, Barrientos, 1999] Pedro L. Galindo, Carmen García López, Juan M. Barrientos Villar. La Cuantificación Vectorial. Escuela Superior de Ingeniería Universidad de Cádiz.1999. Cádiz, España. Pp. 2, 3.**
- ▶ **[Heckbert, 1982] Paul S. Heckbert. Color Image Quantization for Frame Buffer Display. Laboratorio de Computación Gráfica del Intituto de Tecnología de New York. 1982. USA. Pp. 4-8.**

Bibliografía

- ▶ **[Lloyd, 1982] Stuart Lloyd. Least Squares Quantization PCM. IEEE Transactions on Information Theory, Vol. It-28, No. 2, 1982.**
- ▶ **[Lumer y Faieta, 1994]**
- ▶ **E. Lumer y B. Faieta. Diversity and Adaptation in populations of clustering ants. 1994**
- ▶ **[Matteucci, 2008] Clustering: An Introduction. Extraído el 09 de febrero del 2011 desde**
- ▶ **http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/**
- ▶ **[Pelliza, 2007] Transmisión de Datos. Sergio Pelliza. Departamento de Apoyatura Académica. Instituto Salesiano de Estudios Superiores. 2007. México. Pp 8.**

Bibliografía

- ▶ **[Pinto et al., 2005]**
- ▶ **Diego Pinto, Hugo Estigarribia, y Benjamín Barán. Enrutamiento Multicast Multiobjetivos basado en Colonia de Hormigas. Universidad Nacional de Asunción. Centro Nacional de Computación. Asunción, Paraguay. 2005.**
- ▶ **[Piñeiro, 2010] Teoría de la Información y Codificación. Cándido Piñeiro Gómez. 2010. Cap. I, Pp. 2.**
- ▶ **[Ragone y Vittori, 2006] Marcelo Ragone, Juan Pablo Vittori. Fotografía Digital. Especialización en Lenguajes Artísticos Combinados. Instituto Universitario Nacional de Arte. 2006. Buenos Aires, Argentina. Pp 1-7.**
- ▶ **[Ross Dawson, 2007] Future of Media Report. Ross Dawson. Future Exploration Network. 2007. Sydney, Australia. Pp. 6.**

Bibliografía

- ▶ [Seagate, 2008] ¿Qué capacidad de almacenamiento necesitan sus clientes?. Seagate. EE.UU. 2008.
- ▶ [Viktor Macura, Ant Colony Algorithm] Vik tor Makura, Ant Colony Algorithm. Extraído el 09 de febrero del 2011 desde <http://mathworld.wolfram.com/AntColonyAlgorithm.html>

GRACIAS